REMARKS

The Examiner continues to reject claims 1-9 under 35 U.S.C. 102(e) as being anticipated by Schmidt (U.S. Patent No. 6523059). Additionally, the Examiner has rejected claims 1, 4, and 7 under 35 U.S.C. 103(a) as unpatentable over Jeffords et al. (U.S. Patent No. 6510478). These rejections are respectfully traversed as outlined below.

Rejections under 35 U.S.C. 102(e)

Applicant has amended the independent claims to recite, taking claim 1 as an example, "saving a snapshot of a state of the first thread and thereafter setting the state of the first thread to a safe state" and "restoring the state of the first thread from the snapshot." Support for this amendment can be found, for example, in Fig. 1, Fig. 6A and Fig. 6B.

The Schmidt reference does not appear to anticipate or suggest such a feature. For example, at col. 7, lines 38-48, Schmidt discloses:

> In the described embodiment, a global safepoint operation, such as a garbage collection operation, is to be performed for thread 200c. The global safepoint operation is typically expected to be programmed such that the global safepoint operation itself is embodied in code that is considered to be a safe region of code. Accordingly, when thread 200c initiates a "begin safepoint" 300, an assumption may be made that thread 200c is moving through a safe region of code. Thus, it should not be necessary to perform a check regarding the current state of a state flag associated with thread 200c.

On the other hand, as discussed in Applicant's specification (at, for example, page 9, lines 9-17),

> A rendezvous operation is generally arranged to enable the thread to determine its status with respect to a requester thread, e.g., the thread which requested a check point. A rendezvous operation also synchronizes between a reader thread and a writer thread, and handles notification of a writer and blocking of the reader, if necessary. One example of a suitable rendezvous operation will be described below with reference to FIG. 6A and 6B. Once the rendezvous operation performed in step 110 is completed, the cached GC state is restored or otherwise reached with respect to the thread in step 114, and the processing performed by an unsafe thread during a check point is completed.

In summary, no assumption is made about the state of the thread. The present state of the thread is "cached" and the thread is explicitly put into a safe state.

There is no suggestion in Schmidt for such a feature and, in fact, Schmidt explicitly teaches away from such a feature (given its assumption regarding the safe state).

With regard to Jeffords, this reference does not even consider whether a requesting process is in a "safe" state. Rather, a "shared object" is controlled by a "lock owner" and all requests for access to the shared object must be through the lock owner." See, e.g., col. 1, lines

36-50. A requesting process can never access the shared object without permission of the lock owner and, thus, there is no disclosure or suggestion of the requesting process (taking as given, for the sake of argument, that a "process" is a "thread") caching the present state and being explicitly put into a safe state.

For at least these reasons, then, it is respectfully submitted that the subject matter of the claims is patentable over Schmidt and Jeffords.

CONCLUSION

Applicants believe that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Alan S. Hodes
Reg. No. 38,185

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300